

19 RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

11 N° de publication :  
(à n'utiliser que pour les  
commandes de reproduction)

2 687 811

21 N° d'enregistrement national : 93 01908

51 Int Cl<sup>5</sup> : G 06 F 12/02 , G 11 C 16/00

12

## DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 19.02.93.

30 Priorité : 20.02.92 JP 3339892.

43 Date de la mise à disposition du public de la  
demande : 27.08.93 Bulletin 93/34.

56 Liste des documents cités dans le rapport de  
recherche : Le rapport de recherche n'a pas été  
établi à la date de publication de la demande.

60 Références à d'autres documents nationaux  
apparentés :

71 Demandeur(s) : FUJITSU LIMITED — JP.

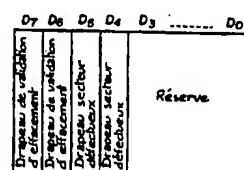
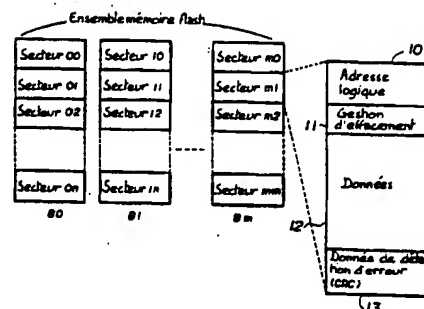
72 Inventeur(s) : Itoh Hiroyuki — c/o Fujitsu Limited, et  
Matsui Noriyuki — c/o Fujitsu Limited.

73 Titulaire(s) :

74 Mandataire : Cabinet Beau de Loménie.

54 Procédé et appareil de commande de mémoire flash.

57 Le procédé de commande de mémoire flash selon l'invention comporte les opérations consistant à (a) diviser la région de mémorisation d'une mémoire flash en plusieurs secteurs, où chaque secteur comporte une partie adresse logique (10) destinée à mémoriser une adresse logique du secteur, une partie gestion d'effacement (11) destinée à emmagasiner des informations qui indiquent au moins si le secteur peut être effacé, et une partie données (12) destinée à emmagasiner des données, et (b) à faire accès à un secteur arbitraire de la mémoire flash en spécifiant l'adresse logique du secteur arbitraire. L'invention concerne également un appareil de commande de mémoire flash qui utilise ce procédé.



FR 2 687 811 - A1



La présente invention concerne de façon générale les procédés et les appareils de commande de mémoire flash et, plus particulièrement, un procédé de commande de mémoire flash qui autorise la mise à jour des données à grande vitesse et un appareil de commande de mémoire flash qui utilise ce procédé de commande de mémoire flash.

Les appareils de traitement de données utilisent comme fichiers des disques tels que des disques magnétiques et des disquettes. Toutefois, ces disques sont relativement lourds en raison de leur structure mécanique et, en outre, ils ont une consommation électrique relativement forte.

C'est pourquoi, on peut concevoir d'utiliser comme unités de mémorisation en mémoire de masse, à la place des disques, des mémoires semi-conductrices volatiles, par exemple des mémoires vives dynamiques (DRAM) et des mémoires vives statiques (SRAM). Toutefois, les mémoires semiconductrices volatiles demandent une source d'alimentation électrique de secours, comme une pile au lithium, et les DRAM et les SRAM ne font pas exception. C'est pourquoi on peut concevoir d'utiliser comme unité de mémorisation en mémoire de masse une mémoire flash qui a été récemment mise au point. La mémoire flash ne demande pas de source d'alimentation électrique de secours et est peu coûteuse si on la compare aux DRAM et aux SRAM.

Cette mémoire flash peu coûteuse est une mémoire non volatile, ou rémanente, et est en outre à réécriture électrique. Toutefois, avec la mémoire flash, les problèmes suivants se posent.

Tout d'abord, on ne peut pas écrire les données avant que les données préalablement emmagasinées n'aient été effacées. Toutefois, l'effacement des données ne peut pas se faire à raison d'un octet à la fois, par exemple. L'effacement des données doit se faire par blocs, ou bien il doit se faire par rapport à la puce de mémoire flash tout entière. Le bloc est une unité relativement grande et peut se monter à une valeur de plusieurs dizaines de kilo-octets à plusieurs centaines de kilo-octets.

En deuxième lieu, il est impossible de réécrire les données à une adresse arbitraire dans l'espace des adresses physiques, puisque, comme indiqué ci-dessus, l'effacement des données doit se faire par blocs. Par conséquent, lors de la réécriture des données, il est nécessaire de sauvegarder temporairement les données contenues dans les blocs en un certain emplacement avant d'effectuer l'effacement des données par blocs. Après cela, on écrit, à partir de l'emplacement

de sauvegarde, les données qui ne doivent pas être effacées, puis on écrit les données nouvelles.

En troisième lieu, puisque la réécriture des données impose l'ennuyeux processus consistant à temporairement sauvegarder les données et à effectuer  
5 l'effacement des données par blocs comme ci-dessus indiqué, la vitesse d'écriture ou la vitesse d'effacement sont considérablement affaiblies par comparaison à la vitesse de lecture. De plus, puisque l'effacement des données s'effectue par blocs relativement grands, la réécriture des données demande un temps extrêmement long pour être menée à son terme. En résultat, dans le cas d'une banque de données  
10 qui traite une grande quantité de données, la réécriture de données est pratiquement impossible.

Par conséquent, la mémoire flash présente un grave problème en ce que la réécriture de données ne peut pas être effectuée par un moyen d'accès qui utilise une adresse de la même façon que la DRAM, la SRAM ou le disque dur. Pour cette  
15 raison, il est pratiquement impossible d'utiliser la mémoire flash à la place de la DRAM, de la SRAM, du disque dur, ou autres. Toutefois, puisque la mémoire flash ne demande pas de source d'alimentation électrique de secours et est peu coûteuse, il existe une demande pour un moyen permettant de tirer parti des particularités avantageuses de la mémoire flash.

20 Un but général de l'invention est donc de produire un procédé et un appareil de commande de mémoire flash dans lesquels les problèmes ci-dessus décrits sont éliminés.

Un autre but, plus particulier, de l'invention est de produire un procédé de commande de mémoire flash comprenant les opérations suivantes : (a) diviser la  
25 région de mémorisation d'une mémoire flash en plusieurs secteurs, où chacun des secteurs comporte une partie adresse logique destinée à emmagasiner une adresse logique du secteur, une partie gestion d'effacement destinée à emmagasiner des informations indiquant au moins si le secteur peut être effacé, et une partie données destinée à emmagasiner des données, et (b) faire accès à un secteur arbitraire de la  
30 mémoire flash en spécifiant l'adresse logique du secteur arbitraire. Selon le procédé de commande de mémoire flash de l'invention, il est possible d'utiliser la mémoire flash peu coûteuse comme fichier ou comme unité de mémorisation en mémoire de masse, parce que les données peuvent être réécrites facilement et à grande vitesse grâce à une gestion de la région de mémorisation de la mémoire flash en plusieurs  
35 secteurs.

Un autre but de l'invention est de produire un appareil de commande de mémoire flash comprenant une mémoire flash qui possède une région de mémorisation divisée en plusieurs secteurs, lesquels comportent chacun une partie adresse logique destinée à emmagasiner une adresse logique du secteur, une partie  
5 gestion d'effacement destinée à emmagasiner des informations indiquant au moins si le secteur peut être effacé et une partie données destinée à emmagasiner des données, et un moyen de commande couplé à la mémoire flash pour faire accès à un secteur arbitraire de la mémoire par spécification de l'adresse logique du secteur arbitraire. Selon l'appareil de commande de mémoire flash de l'invention, il est  
10 possible d'utiliser la mémoire flash peu coûteuse comme fichier ou unité de mémorisation en mémoire de masse, parce que les données peuvent être réécrites facilement et à grande vitesse grâce à une gestion de la région de mémorisation de la mémoire flash en plusieurs secteurs.

La description suivante, conçue à titre d'illustration de l'invention, vise  
15 à donner une meilleure compréhension de ses caractéristiques et avantages ; elle s'appuie sur les dessins annexés, parmi lesquels :

- la figure 1 est un schéma permettant d'expliquer le principe de fonctionnement de l'invention ;
- la figure 2 est un schéma fonctionnel d'ensemble montrant un mode  
20 de réalisation d'un appareil de commande de mémoire flash selon l'invention ;
- la figure 3 est un schéma servant à expliquer l'état d'utilisation d'une mémoire flash du mode de réalisation représenté sur la figure 2 ;
- la figure 4 est un organigramme servant à expliquer un processus d'écriture du mode de réalisation ; et
- 25 - la figure 5 est un organigramme servant à expliquer un processus de lecture du mode de réalisation.

On va tout d'abord décrire le principe de fonctionnement de l'invention, en se reportant à la figure 1.

Selon l'invention, une mémoire flash est gérée en plusieurs secteurs 00  
30 à 0n. Les secteurs possèdent tous la même structure, et la figure 1 montre la structure du secteur 02 à titre d'exemple. Comme représenté, le secteur 02 comporte une partie adresse logique 10, une partie gestion d'effacement 11, une partie données 12, et une partie données de détection d'erreur 13.

Une adresse de secteur, qui est une adresse logique, est attribuée à  
35 chaque secteur, de sorte que l'accès aux données s'effectue à l'aide de l'adresse du secteur, et non d'une adresse physique. Si une instruction d'écriture est délivrée, des

bits de gestion d'effacement contenus dans la partie gestion d'effacement 11 du secteur considéré sont positionnés de façon à indiquer que la partie données 12 de ce secteur a été invalidée. De plus, les données devant être nouvellement écrites sont écrites dans le secteur vide de la mémoire flash, et l'adresse logique de ce secteur est fixée à une adresse logique qui est identique à celle du secteur qui a été invalidé par le positionnement des bits de gestion d'effacement.

Dans une région de la mémoire flash où une donnée a déjà été écrite, on ne peut pas réécrire une donnée différentes en spécifiant l'adresse. Toutefois, la donnée peut être écrite dans une région vide où aucune donnée n'a été écrite, c'est-à-dire une région qui se trouve dans un état initial, par spécification de l'adresse. L'invention utilise cette caractéristique de la mémoire flash de façon à obtenir des effets qui seraient identiques à ceux obtenus si les données étaient sensiblement écrites par secteurs. En d'autres termes, les effets procurés par l'invention sont équivalents à ceux qu'on obtiendrait si la réécriture des données avait été effectuée par secteurs.

Lorsque les données sont écrites, dans chaque secteur, la partie gestion d'effacement 11 de celui-ci est repositionnée à l'état initial. Ensuite, une adresse logique est écrite dans la partie adresse logique 10 du secteur, et les données qui doivent être écrites à cette adresse logique sont écrites dans la partie données 12 du secteur. De plus, une donnée de détection d'erreur, par exemple un code de contrôle de redondance cyclique (SRC) est écrit dans la partie donnée de détection d'erreur 13 du secteur. Par exemple, la donnée de détection d'erreur autorise la correction de l'erreur dans 1 bit et la détection de l'erreur dans 2 bits.

Si la donnée du secteur 02 doit être réécrite, en relation avec la figure 1, et que des secteurs vides commencent à partir du secteur 0n-1, les bits de gestion d'effacement contenus dans la partie gestion d'effacement 11 du secteur 02 sont positionnés par écriture de données complémentaires à celles existant au moment de l'état initial. Ensuite, une adresse logique qui est identique à celle du secteur 02 est écrite dans la partie adresse logique 10 du secteur 0n-1, et la nouvelle donnée est écrite dans la partie donnée 12 du secteur 0n-1. De plus, la donnée correcte de détection et de correction d'erreur est écrite dans la partie donnée de détection d'erreur 13 du secteur 0n-1. Dans ce cas, la partie gestion d'effacement 11 du secteur 02, la partie adresse logique 10 du secteur 0n-1 et la partie données 12 du secteur 0n-1 sont toutes dans l'état initial avant que l'écriture ait lieu, de sorte que l'opération d'écriture se limite aux régions se trouvant dans l'état initial et peut être effectuée avec une vitesse élevée.

Lors de la lecture réelle des données dans la région d'adresse logique, les parties adresse logique 10 de chacun des secteurs sont successivement balayées. Tout d'abord, la destination de l'accès est détectée à partir de la partie adresse logique 10 du secteur 02. Toutefois, puisque les bits de gestion d'effacement de la partie gestion d'effacement 11 du secteur 02 sont positionnés, il est reconnu que le secteur 02 a été effacé, et le balayage des parties adresse logique 10 des secteurs restants continue. L'adresse logique de la destination de l'accès est détectée à partir de la partie adresse logique 10 du secteur 0n-1, et les données nécessaires sont obtenues dans la partie données 12 de ce secteur 0n-1.

Par conséquent, la réécriture des données peut être effectuée à grande vitesse à l'aide de la mémoire flash, sans qu'il soit nécessaire d'effacer réellement la mémoire flash, puis de réécrire ensuite les nouvelles données.

On va maintenant décrire un mode de réalisation d'un appareil de commande de mémoire flash selon l'invention, en se reportant aux figures 2 et 3. La figure 2 représente une partie essentielle de ce mode de réalisation, et la figure 3 est un schéma servant à expliquer l'état d'utilisation de la mémoire flash selon ce mode de réalisation. Ce mode de réalisation de l'appareil emploie un mode de réalisation d'un procédé de commande de mémoire flash selon l'invention, comme décrit ci-après, plus particulièrement en relation avec les figures 4 et 5 qui seront décrites ultérieurement.

Sur la figure 2, on voit que l'appareil de commande de mémoire flash comporte de façon générale un microprocesseur (MPU) 1, une mémoire flash 2, et une unité d'entrée-sortie (I/O) 3, qui sont connectés par l'intermédiaire d'un bus 5. La MPU 1 commande l'accès relatif à la mémoire flash 2, et comporte une partie 4 de gestion de secteurs vides, qui gère les secteurs vides de la mémoire flash 2. La mémoire flash possède la structure représentée sur la figure 3 par exemple. L'unité d'entrée-sortie 3 reçoit en entrée les données qui doivent être emmagasinées dans la mémoire flash 2 et délivre en sortie les données nécessaires qui ont été lues dans la mémoire flash 2.

Comme on peut le voir dans la partie (A) de la figure 3, la mémoire flash 2 est constituée de blocs B0 à Bm. Le bloc B0 est constitué de secteurs 00 à 0n, le bloc B1 est constitué de secteurs 10 à 1n, ..., et le bloc Bm est constitué de secteurs m0 à mn.

Chaque bloc correspond à l'unité minimale sur laquelle on peut procéder à l'effacement, c'est-à-dire où l'effacement peut être un effacement du type bloc. Chaque bloc peut correspondre à une puce de mémoire flash ou à une

partie de la puce de mémoire flash. Par conséquent, les blocs B0 à Bm représentés dans la partie (A) de la figure 3 peuvent correspondre à une ou plusieurs puces de mémoire flash, et la mémoire flash 2 représentée sur la figure 2 peut être constituée d'une ou plusieurs puces de mémoire flash.

5 Les secteurs formant chaque bloc respectif ont une taille arbitraire qui est choisie de façon appropriée. Comme décrit ci-dessus en relation avec la figure 1, chaque secteur comporte une partie adresse logique 10, une partie gestion d'effacement 11, une partie données 12, et une partie donnée de détection d'erreur 13. Lorsqu'on utilise la mémoire flash 2 comme fichier par exemple, les données  
10 qui doivent être emmagasinées dans le fichier sont écrites dans la partie données 12 du secteur.

Comme représenté dans la partie (B) de la figure 3, un drapeau de validation d'effacement qui indique que le secteur considéré a été effacé et est invalidé est écrit aux bits D7 et D6 de la partie gestion d'effacement 11 de ce  
15 secteur considéré. De plus, un drapeau secteur défectueux qui indique que le secteur considéré est défectueux est écrit aux bits D5 et D4 de la partie gestion d'effacement 11 de ce secteur considéré. Dans ce mode de réalisation, le drapeau de validation d'effacement et le drapeau secteur défectueux sont écrits en groupes de données doubles afin d'améliorer la fiabilité de l'appareil de commande de  
20 mémoire flash, mais il est naturellement possible de prévoir plus de deux drapeaux de validation d'effacement et plus de deux drapeaux secteur défectueux. Les bits restants, soit D3 à D0, de la partie gestion d'effacement 11 sont des bits de réserve.

Comme on peut le voir sur la figure 3, si des données sont écrites jusqu'au secteur m0, la partie 4 de gestion des secteurs vides de la MPU 1 représentée sur la figure 2 gère des informations indiquant que les secteurs vides  
25 commencent à partir du secteur m1. Si une demande d'accès pour lecture est faite depuis l'unité d'entrée-sortie 3 à destination de la mémoire flash 2 via la MPU 1, par exemple, la MPU 1 balaye successivement les adresses logiques de chacun des secteurs compris à l'intérieur de l'ensemble mémoire flash à partir du secteur 00 de la partie (A) de la figure 3. Lorsque l'adresse logique d'un secteur dans lequel le  
30 drapeau de gestion d'effacement n'est pas positionné concorde avec l'adresse logique de la destination de l'accès, les données de ce secteur sont lues et transmises à l'unité d'entrée-sortie 3.

D'autre part, si une demande d'accès pour réécriture est faite depuis  
35 l'unité d'entrée-sortie 3 à destination de la mémoire flash 2, la MPU 1 détermine dans la mémoire flash 2 un secteur qui possède l'adresse logique de la destination

d'accès et positionne le drapeau de validation d'effacement aux bits D7 et D6 de la partie gestion d'effacement 11 de ce secteur. La mémoire flash 2 peut mémoriser la donnée "0" ou "1" dans l'état initial. Ainsi, on positionne le drapeau de validation d'effacement aux bits D7 et D6 de la partie gestion d'effacement 11 en mettant en place des "1" si les bits D7 et D6 emmagasinent initialement des "0" et en mettant en place des "0" si les bits D7 et D6 emmagasinent initialement des "1".

Après cela, la MPU 1 retrouve dans la partie 4 de gestion des secteurs vides le secteur à partir duquel les secteurs vides commencent. Par exemple, la MPU 1 détermine le secteur m1 comme étant le secteur à partir duquel les secteurs vides commencent, et fait accès au drapeau secteur défectueux des bits D5 et D4 dans la partie gestion d'effacement 11 du secteur m1. Si le drapeau secteur défectueux n'est pas positionné dans la partie gestion d'effacement 11 du secteur m1, la MPU 1 écrit l'adresse logique de la destination d'accès dans la partie adresse logique 10, les nouvelles données à écrire dans la partie données 12, et la donnée de détection d'erreur dans la partie donnée de détection d'erreur 13 du secteur m1, et mémorise dans la partie 4 de gestion des secteurs vides des informations indiquant que le prochain secteur vide est le secteur m2.

Toutefois, si l'un des bits D5 et D4 du drapeau secteur défectueux est positionné dans la partie gestion d'effacement 11 du secteur m1, la MPU 1 effectuera l'opération d'écriture relativement au secteur m2 suivant, de la façon ci-dessus indiquée.

On va ensuite décrire de façon détaillée l'opération d'écriture selon ce mode de réalisation, en relation avec la figure 4. La figure 4 est un organigramme montrant l'opération d'écriture de la MPU 1 de la figure 2.

Comme on peut le voir sur la figure 4, l'étape S1 produit l'introduction d'une adresse logique d'entrée en provenance de l'unité d'entrée-sortie 3 de la figure 2, et l'étape S2 produit le balayage des secteurs de la mémoire flash 2 dans lesquels des données sont déjà écrites. A l'étape S3, il est déterminé si la mémoire flash 2 possède un secteur ayant la même adresse logique que l'adresse logique d'entrée. L'organigramme passe à l'étape S4 en cas de réponse positive à l'étape S3, mais avance à l'étape S6 en cas de réponse négative à l'étape S3.

A l'étape S4, il est déterminé si le secteur possédant la même adresse logique que l'adresse logique d'entrée possède une partie gestion d'effacement 11 dans laquelle le drapeau de validation d'effacement est positionné. L'organigramme revient à l'étape S2 en cas de réponse positive à l'étape S4. Inversement, si la réponse est négative à l'étape S4, alors, à l'étape S5, le drapeau de validation



d'effacement de la partie gestion d'effacement 11 de ce secteur est positionné, et, à l'étape S6, la donnée qui doit être écrite est introduite depuis l'unité d'entrée-sortie 3. Ensuite, à l'étape S7, est effectuée la recherche de la région de la mémoire flash 2 dans laquelle la donnée doit être écrite.

5 Dans ce cas, la mémoire flash 2 possède une capacité de mémorisation totale atteignant  $m-1$  blocs même si  $m$  blocs sont prévus, et un bloc est utilisé comme bloc de travail. Comme décrit ci-dessus, chaque bloc peut correspondre à une ou plusieurs puces de mémoire flash. Ainsi, parmi les  $m$  blocs de la mémoire flash 2, seuls  $m-1$  blocs sont utilisés au titre de la région de mémorisation effective  
10 de la mémoire flash 2.

A l'étape S8, il est déterminé si la mémoire flash 2 comporte ou non une région vide mise à part le bloc de travail ci-dessus indiqué. L'organigramme passe à l'étape S9 en cas de réponse négative à l'étape S8, et passe à l'étape S12 en cas de réponse positive.

15 A l'étape S9, est recherché le bloc possédant le plus grand nombre possible de drapeaux de validation d'effacement positionnés, dans les secteurs des blocs. Ensuite, à l'étape S10, est effectuée la sauvegarde, dans le bloc de travail, de ceux des secteurs formant le bloc trouvé à l'étape S9 dans lesquels le drapeau de validation d'effacement n'est pas positionné, par copie du contenu des secteurs  
20 dans le bloc de travail. Après cela, à l'étape S11, le bloc trouvé à l'étape S9 est effacé et est alors considéré comme le nouveau bloc de travail.

L'étape S12 vise à confirmer que le drapeau secteur défectueux du secteur qui est placé au début de la région vide n'est pas positionné et passe au secteur suivant si le drapeau secteur défectueux est positionné. A l'étape S13,  
25 l'adresse, les données et la donnée de détection d'erreur sont écrites dans la région vide.

A l'étape S14, il est déterminé si l'opération d'écriture a été ou non réussie. L'organigramme prend fin en cas de réponse positive à l'étape S14. Inversement, en cas de réponse négative, il est procédé, à l'étape S15, au positionnement  
30 du drapeau secteur défectueux pour le secteur par rapport auquel l'opération d'écriture n'a pas réussi, puis l'organigramme revient à l'étape S8.

Une fois que l'opération d'écriture relative à la région de mémorisation, c'est-à-dire aux  $m-1$  blocs, a pris fin, il est possible de réaliser un ajustement des données à l'aide du bloc de travail. Par exemple, le bloc (ou puce de mémoire  
35 flash) qui possède la plus grande région effaçable est déterminé parmi les blocs (ou la puce ou bien les puces de mémoire flash) dans lesquels des données ont déjà été

écrites. Dans le bloc qui possède la plus grande région effaçable, la région qui ne doit pas être effacée est sauvegardée dans le bloc de travail par copie de son contenu, et la donnée devant être écrite ensuite est écrite après ce contenu. En outre, le bloc ci-dessus possédant la plus grande région effaçable est effacé après  
5 que le contenu à sauvegarder a été copié dans le bloc de travail, et le bloc effacé est préparé pour être utilisé comme nouveau bloc de travail pendant l'ajustement de données suivant.

On va maintenant décrire de façon détaillée l'opération de lecture selon ce mode de réalisation. La figure 5 est un organigramme montrant l'opération de  
10 lecture de la MPU 1 représentée sur la figure 2.

Comme on peut le voir sur la figure 5, l'étape S21 produit l'introduction d'une adresse logique d'entrée en provenance de l'unité d'entrée-sortie 3 de la figure 2, et l'étape S22 effectue le balayage des secteurs de la mémoire flash 2 dans  
15 lesquels des données sont déjà écrites. A l'étape S23, il est déterminé si la mémoire flash 2 possède un secteur ayant la même adresse logique que l'adresse logique d'entrée. L'organigramme passe à l'étape S25 en cas de réponse positive à l'étape S23, mais va à l'étape S24 en cas de réponse négative à l'étape S23. A l'étape S24, une erreur de lecture est indiquée à l'unité d'entrée-sortie 3 via la MPU 1.

A l'étape S25, il est décidé si le secteur ayant la même adresse logique  
20 que l'adresse logique d'entrée possède une partie gestion d'effacement 11 dans laquelle le drapeau de validation de validation d'effacement est positionné. L'organigramme revient à l'étape S22 en cas de réponse positive à l'étape S25. Inversement, en cas de réponse négative à l'étape S25, alors, à l'étape S26, sont lues les données du secteur ayant la même adresse logique que l'adresse logique d'entrée,  
25 et, à l'étape S27, les données lues sont transmises à l'unité d'entrée-sortie 3.

A l'étape S28, est produite une donnée de détection d'erreur qui est associée à la donnée lue et indique l'erreur qui s'y trouve. A l'étape S29, la donnée de détection d'erreur qui est produite à l'étape S28 est comparée avec la donnée de détection d'erreur venant de la partie donnée de détection d'erreur 13 du secteur. A  
30 l'étape S30, il est déterminé si les deux données de détection d'erreur comparées concordent ou non. L'organigramme arrive à sa fin en cas de réponse positive à l'étape S30, mais passe à l'étape S31 en cas de réponse négative. A l'étape S31, une erreur de lecture est indiquée à l'unité d'entrée-sortie 3 via la MPU 1.

De plus, à l'étape S32, il est déterminé si le drapeau secteur défectueux  
35 contenu dans la partie de gestion d'effacement 11 du secteur est ou non positionné. L'organigramme arrive à sa fin en cas de réponse positive à l'étape S32. Inverse-

ment, en cas de réponse négative, alors, à l'étape S33, le drapeau secteur défectueux de la partie gestion d'effacement du secteur est positionné, puis l'organigramme prend fin.

- 5 Bien entendu, l'homme de l'art sera en mesure d'imaginer, à partir du procédé et de l'appareil dont la description vient d'être donnée à titre simplement illustratif et nullement limitatif, diverses variantes et modifications ne sortant pas du cadre de l'invention.

## REVENDICATIONS

1. Procédé de commande de mémoire flash, caractérisé en ce que sont prévues les opérations suivantes :

- 5 (a) diviser une région de mémorisation d'une mémoire flash (2) en plusieurs secteurs, contenant chacun une partie adresse logique (10) destinée à emmagasiner une adresse logique du secteur, une partie gestion d'effacement (11) destinée à emmagasiner des informations indiquant au moins si le secteur peut être effacé, et une partie données (12) destinée à emmagasiner des données ; et
- 10 (b) faire accès à un secteur arbitraire de la mémoire flash (2) en spécifiant l'adresse logique du secteur arbitraire.

2. Procédé selon la revendication 1, caractérisé en ce que la région de mémorisation de la mémoire flash (2) est une région effaçable du type par blocs, où chaque bloc est l'unité minimale pour laquelle l'effacement de la mémoire flash

15 peut être fait, et ladite opération (a) divise la région de mémorisation de façon que chaque bloc soit constitué par plusieurs secteurs.

3. Procédé selon la revendication 2, caractérisé en ce que la mémoire flash (2) est constituée d'une ou plusieurs puces de mémoire flash.

4. Procédé selon la revendication 2, caractérisé en ce que la région de

20 mémorisation totale de la mémoire flash (2) se monte à  $m$  blocs, et  $m-1$  blocs sont utilisés au titre de région de mémorisation effective, tandis que le bloc restant est utilisé comme bloc de travail.

5. Procédé selon la revendication 4, caractérisé en ce que l'opération suivante est en outre prévue :

- 25 (c) ajuster à l'aide du bloc de travail les données qui ont été écrites dans la mémoire flash (2) après qu'une opération d'écriture relative à tous les  $m-1$  blocs a été menée à son terme.

6. Procédé selon la revendication 1, caractérisé en ce que l'opération (a) divise la région de mémorisation de façon que chaque secteur comporte en outre

30 une partie donnée de détection d'erreur (13) destinée à emmagasiner des informations qui sont utilisées pour détecter et corriger une erreur dans les données emmagasinées dans la partie données (12).

7. Procédé selon la revendication 1, caractérisé en ce que l'opération (a) divise la région de mémorisation de façon que la partie gestion d'effacement (11)

35 de chaque secteur comporte un drapeau de validation d'effacement qui possède un état initial si la partie données (12) de ce même secteur est en cours de validité et

possède un état autre que l'état initial si la partie données de ce secteur a été invalidée.

8. Procédé selon la revendication 7, caractérisé en ce que la partie gestion d'effacement (11) de chaque secteur comporte plusieurs drapeaux de validation d'effacement.

9. Procédé selon la revendication 1, caractérisé en ce que l'opération (a) divise la région de mémorisation de façon que la partie gestion d'effacement (11) de chaque secteur comporte un drapeau secteur défectueux qui indique si le secteur est ou non défectueux.

10. Procédé selon la revendication 9, caractérisé en ce que la partie gestion d'effacement (11) de chaque secteur comporte plusieurs drapeaux secteur défectueux.

11. Procédé selon la revendication 1, caractérisé en ce que l'opération suivante est en outre prévue :

(c) gérer les secteurs vides de la mémoire flash (2) sur la base des informations emmagasinées dans la partie gestion d'effacement (11) de chaque secteur.

12. Appareil de commande de mémoire flash, caractérisé en ce que sont prévus :

une mémoire flash (2) possédant une région de mémorisation qui est divisée en plusieurs secteurs comportant chacun une partie adresse logique (10) destinée à emmagasiner une adresse logique du secteur, une partie gestion d'effacement (11) servant à emmagasiner des informations qui indiquent au moins si le secteur peut être effacé, et une partie données (12) destinée à emmagasiner des données ; et

un moyen de commande (1) couplé à ladite mémoire flash afin de faire accès à un secteur arbitraire de ladite mémoire (2) par spécification de l'adresse logique du secteur arbitraire.

13. Appareil selon la revendication 12, caractérisé en ce que la région de mémorisation de ladite mémoire flash (2) est une région effaçable du type par blocs, chaque bloc est l'unité minimale pour laquelle un effacement de la mémoire flash (2) peut avoir lieu, et chaque bloc est constitué de plusieurs secteurs.

14. Appareil selon la revendication 13, caractérisé en ce que ladite mémoire flash (2) est constituée d'une ou plusieurs puces de mémoire flash.

15. Appareil selon la revendication 13, caractérisé en ce que la région de mémorisation totale de ladite mémoire flash (2) se monte à m blocs, et  $m-1$

blocs sont utilisés au titre de région de mémorisation effective, tandis que le bloc restant forme un bloc de travail.

16. Appareil selon la revendication 15, caractérisé en ce que ledit moyen de commande (1) comporte un moyen servant à ajuster à l'aide du bloc de travail les données qui ont été écrites dans ladite mémoire flash (2) après qu'une opération d'écriture relative à tous les  $m-1$  blocs a été menée à son terme.

17. Appareil selon la revendication 12, caractérisé en ce que chaque secteur comporte en outre une partie donnée de détection d'erreur (13) destinée à emmagasiner des informations qui sont utilisées pour détecter et corriger une erreur dans les données emmagasinées dans la partie données (12).

18. Appareil selon la revendication 12, caractérisé en ce que la partie gestion d'effacement (11) de chaque secteur comporte un drapeau de validation d'effacement qui possède un état initial si la partie données de ce même secteur est en cours de validité et possède un état autre que l'état initial si la partie données de ce secteur est invalidée.

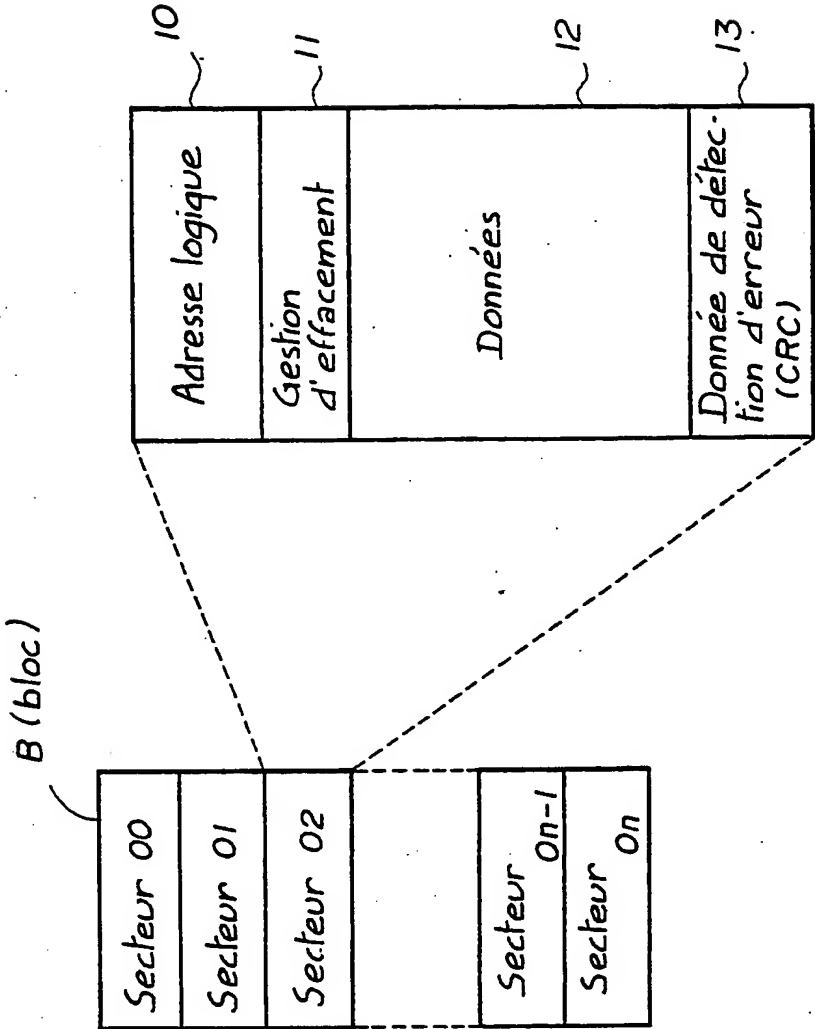
19. Appareil selon la revendication 18, caractérisé en ce que la partie gestion d'effacement (11) de chaque secteur comporte plusieurs drapeaux de validation d'effacement.

20. Appareil selon la revendication 12, caractérisé en ce que la partie gestion d'effacement (11) de chaque secteur comporte un drapeau secteur défectueux qui indique si le secteur est ou non défectueux.

21. Appareil selon la revendication 20, caractérisé en ce que la partie gestion d'effacement (11) de chaque secteur comporte plusieurs drapeaux secteur défectueux.

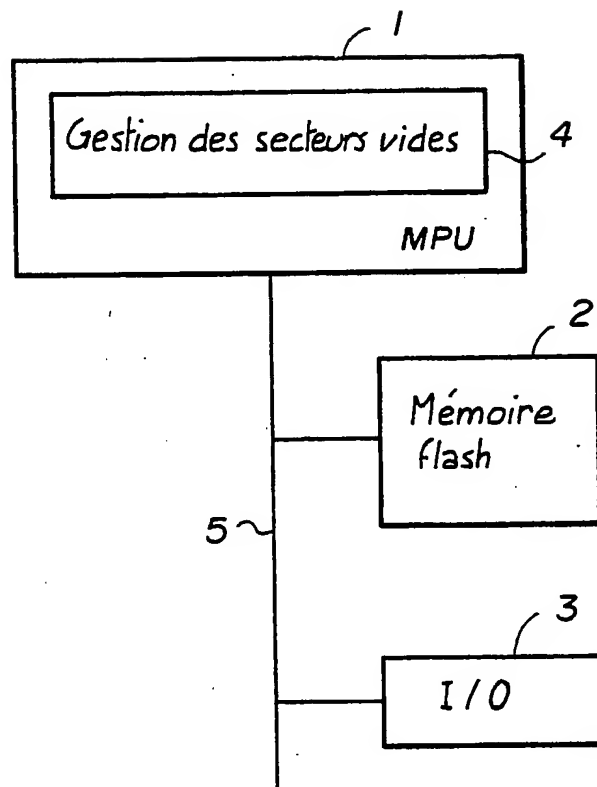
22. Appareil selon la revendication 12, caractérisé en ce que ledit moyen de commande (1) comporte un moyen (4) servant à gérer les secteurs vides de ladite mémoire flash (2) sur la base des informations emmagasinées dans la partie gestion d'effacement (11) de chaque secteur.

FIG. 1



**THIS PAGE BLANK (USPTO)**



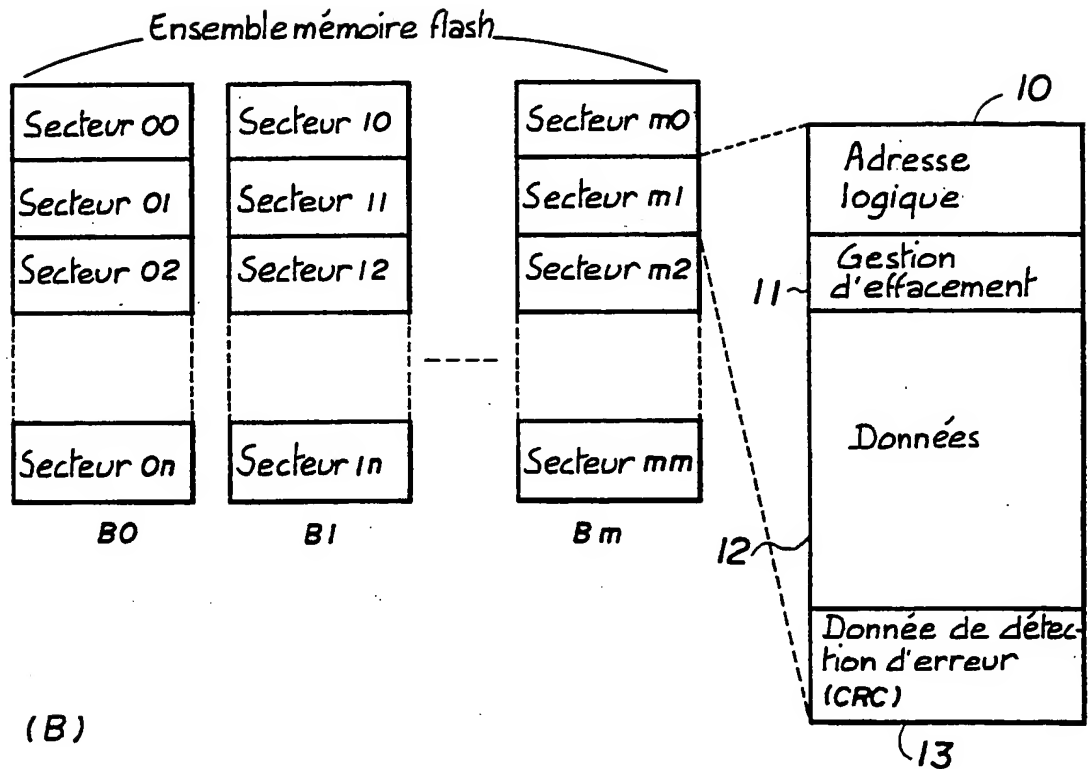
**FIG.2**

**THIS PAGE BLANK (USPTO)**

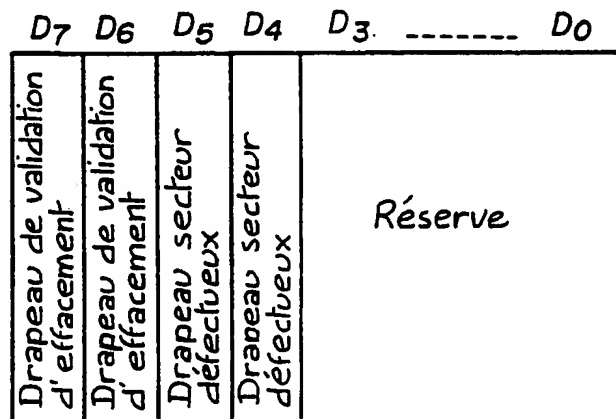
3/5

FIG. 3

(A)



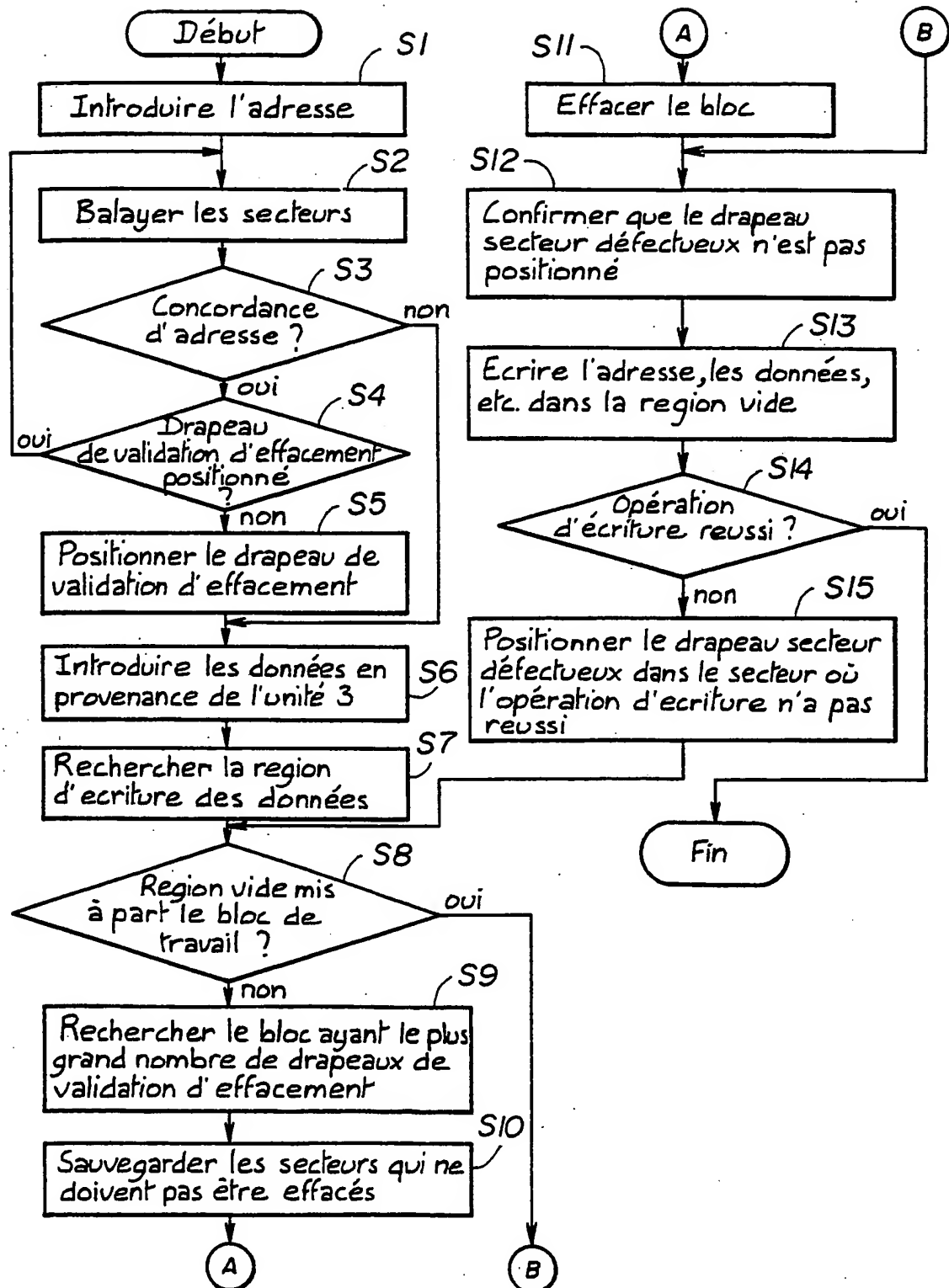
(B)



**THIS PAGE BLANK (USPTO)**

4 / 5

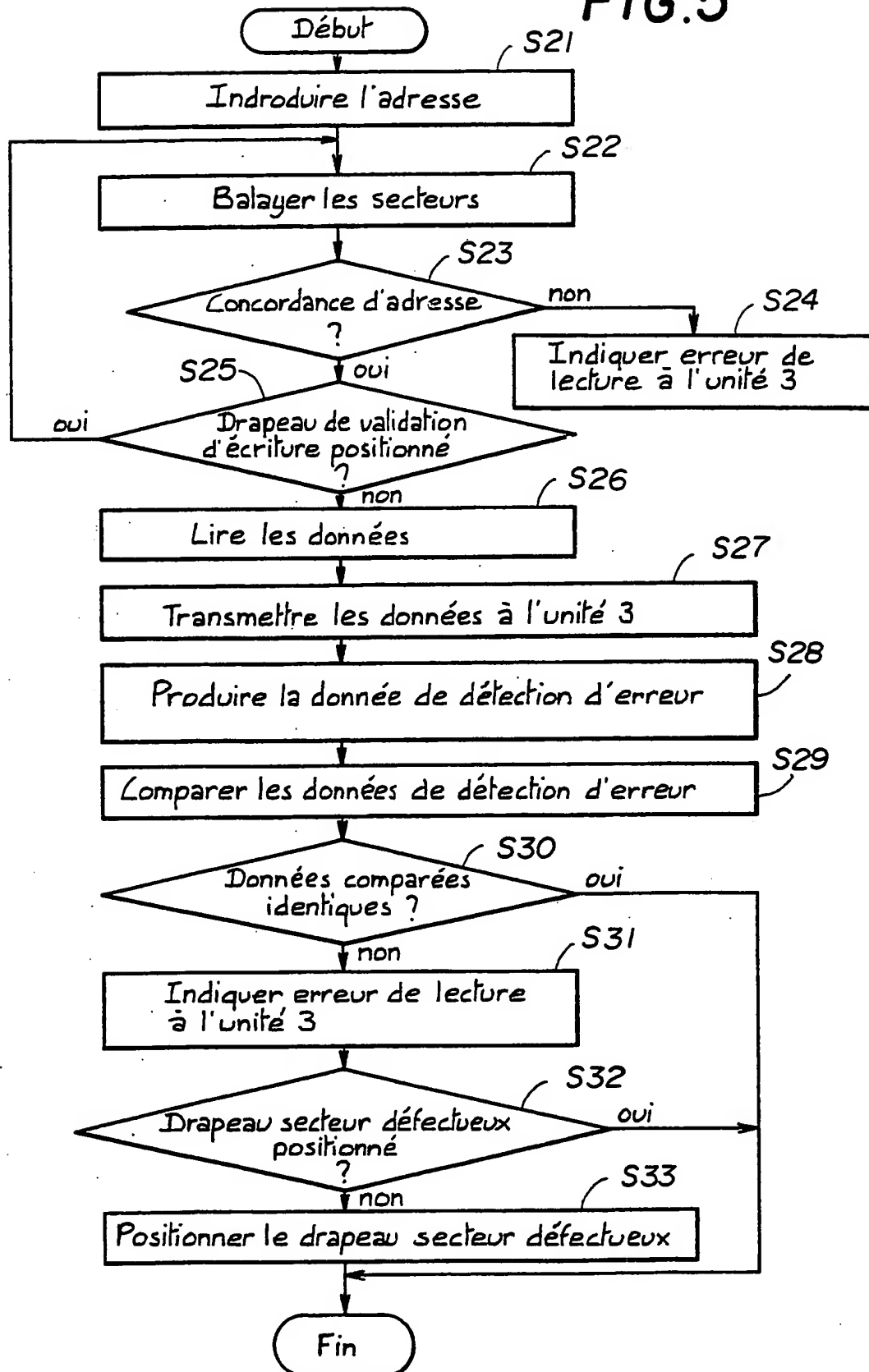
FIG.4



**THIS PAGE BLANK (USPTO)**

5 / 5

FIG.5



**THIS PAGE BLANK (USPTO)**



FR 93 01908

### Process and apparatus for flash memory control

The present invention relates generally to processes and apparatus for flash memory control and, more particularly, to a process for flash memory control which allows the updating of data at high speed and to an apparatus for flash memory control which employs this process for flash memory control.

Data processing apparatus employ disks such as magnetic disks and diskettes as files. However, these disks are relatively heavy owing to their mechanical structure and, furthermore, they have a relatively high consumption of electricity.

This is why semi-conductive volatile memories, for example dynamic random-access memories (DRAM) and static random-access memories (SRAM), can be considered for use as storage units with backing storage instead of disks. However, volatile semi-conductive memories require a standby supply of electricity such as a lithium battery, and CRAM and SRAM are no exception. This is why a flash memory which has recently been developed can be considered for use as a storage unit with backing storage. The flash memory does not require a standby electricity supply and is inexpensive in comparison with DRAM and SRAM.

This inexpensive flash memory is a non-volatile or residual memory and is also adapted for electrical rewriting. However, the following problems arise with the flash memory.

Firstly, data cannot be written before the previously stored data have been deleted. However, deletion of the data cannot be carried out at a rate of one byte at a time, for example. Data deletion has to take place in blocks or has to take place relative to the entire flash memory chip. A block is a relatively large unit and can have a value ranging from several tens of kilo bytes to several hundreds of kilo bytes.

**THIS PAGE BLANK (USPTO)**

Secondly, it is impossible to rewrite data to an arbitrary address in the physical address space since data deletion has to be carried out in blocks, as mentioned above. Consequently, when data are rewritten it is necessary temporarily to save the data contained in the blocks in a certain location before deleting data by blocks. After that, the data which are not to be deleted then the new data are written from the backup location.

Thirdly, since the rewriting of data involves a tedious procedure of temporarily saving data and deleting data in blocks as mentioned above, the speed of writing or the speed of deletion are considerably reduced in comparison with the speed of reading. Furthermore, as data is deleted in relatively great blocks, the rewriting of data takes an extremely long time to complete. As a result, in the case of a data bank which processes a large quantity of data, it is virtually impossible to rewrite data.

Consequently, the flash memory has a serious problem in that data cannot be rewritten by a means of access employing an address in the same way as the DRAM, the SRAM or the hard disk. For this reason, it is virtually impossible to employ the flash memory instead of the DRAM, the SRAM, the hard disk or the like. However, since the flash memory does not require a standby electricity supply and is inexpensive, there is a demand for a means of benefiting from the advantageous features of the flash memory.

A general object of the invention is therefore to provide a process and an apparatus for flash memory control in which the above-described problems are eliminated.

A further more particular object of the invention is to provide a process for flash memory control involving the following operations: (a) dividing the storage region of a flash memory into several sectors where each of the sectors

**THIS PAGE BLANK (USPTO)**

comprises a logic address portion intended to store a logic address of the sector, a deletion management portion intended to store information indicating at least whether the sector can be deleted, and a data portion intended to store data and (b) having access to an arbitrary sector of the flash memory whilst specifying the logic address of the arbitrary sector. With the process according to the invention for flash memory control it is possible to use the inexpensive flash memory as a file or storage unit with backing storage because the data can be rewritten easily and at high speed owing to management of the storage region of the flash memory in several sectors.

A further object of the invention is to produce an apparatus for flash memory control comprising a flash memory having a storage region divided into several sectors which each contain a logic address portion intended to store a logic address of the sector, a deletion management portion intended to store information indicating at least whether the sector can be deleted and a data portion intended to store data, and a control means coupled to the flash memory to give access to an arbitrary sector of the memory by specification of the logic address of the arbitrary sector. With the apparatus for flash memory control according to the invention it is possible to use the inexpensive flash memory as a file or storage unit with backing storage because the data can be rewritten easily and at high speed owing to management of the storage region of the flash memory in several sectors.

The following description, intended to illustrate the invention, aims to allow better understanding of its characteristics and advantages. It relates to the accompanying drawings.

Figure 1 is a diagram illustrating the mode of operation of the invention.

**THIS PAGE BLANK (USPTO)**

Figure 2 is an overall functional diagram showing an embodiment of an apparatus for flash memory control according to the invention.

Figure 3 is a diagram illustrating the state of use of a flash memory of the embodiment shown in figure 2.

Figure 4 is a flow chart illustrating a writing procedure of the embodiment.

Figure 5 is a flow chart illustrating a reading procedure of the embodiment.

The mode of operation of the invention will firstly be described with reference to figure 1.

According to the invention, a flash memory is managed in several sectors 00 to 0n. The sectors all have the same structure, and figure 1 shows the structure of the sector 02 as an example. As shown, the sector 02 comprises a logic address portion 10, a deletion management portion 11, a data portion 12 and an error detecting data portion 13.

A sector address, which is a logic address, is attributed to each sector so access to data can be gained by means of the address of the sector and not a physical address. If a write instruction is delivered, deletion management bits contained in the deletion management portion 11 of the sector under consideration are positioned so as to indicate that the data portion 12 of this sector has been invalidated. Furthermore, the data which are to be rewritten are written in the empty sector of the flash memory and the logic address of this sector is fixed at a logic address identical to that of the sector invalidated by the positioning of the deletion management bits.

In a region of the flash memory where a datum has already been written, it is not possible to rewrite a different datum by

**THIS PAGE BLANK (USPTO)**



specifying the address. However, the datum can be written in an empty region where no datum has been written, that is a region which is in an original state, by specification of the address. The invention employs this characteristic of the flash memory in order to achieve effects which would be identical to those achieved if the data were substantially written by sectors. In other words, the effects achieved by the invention are equivalent to those which would be obtained if data had been rewritten by sectors.

When the data are written in each sector, the deletion management portion 11 thereof is repositioned to the original state. A logic address is then written in the logic address portion 10 of the sector and the data to be written to this logic address are written in the data portion 12 of the sector. Furthermore, an error detection datum, for example a cyclic redundancy control code (SRC), is written in the error detection data portion 13 of the sector. For example, the error detection datum allows correction of the error in one bit and error detection in two bits.

If the datum from sector 02 is to be rewritten, with reference to figure 1, and empty sectors begin from sector 0n-1, the deletion management bits contained in the deletion management portion 11 of the sector 02 are positioned by the writing of data complementary to those existing at the time of the original state. A logic address identical to that of the sector 02 is then written in the logic address portion 10 of the sector 0n-1, and the new datum is written in the data portion 12 of the sector 0n-1. Furthermore, the correct detection and error correction datum is written in the error detection data portion 13 of the sector 0n-1. In this case, the deletion management portion 11 of the sector 02, the logic address portion 10 of the sector 0n-1 and the data portion 12 of the sector 0n-1 are all in the original state before writing takes place so the writing operation is limited to the regions in the original state and can be carried out at high speed.

**THIS PAGE BLANK (USPTO)**

The logic address portions 10 of each of the sectors are scanned successively during the actual reading of data in the logic address region. The destination of access is firstly detected from the logic address portion 10 of the sector 02. However, since the deletion management bits of the deletion management portion 11 of the sector 02 are positioned, it is acknowledged that the sector 02 has been deleted, and scanning of the logic address portions 10 of the remaining sectors continues. The logic address of the destination of access is detected from the logic address portion 10 of the sector 0n-1, and the necessary data are obtained in the data portion 12 of this sector 0n-1.

Consequently, data can be rewritten at high speed by means of the flash memory without the need actually to delete the flash memory then to rewrite the new data.

An embodiment of an apparatus for flash memory control according to the invention will now be described with reference to figures 2 and 3. Figure 2 shows an essential portion of this embodiment, and figure 3 is a diagram illustrating the state of use of the flash memory according to this embodiment. This embodiment of the apparatus employs an embodiment of a process for flash memory control according to the invention, as described hereinafter, more particularly with reference to figures 4 and 5 which will be described at a later stage.

In figure 2 it can be seen that the apparatus for flash memory control generally comprises a microprocessor (MPU) 1, a flash memory 2 and an input-output unit (I/O) 3 which are connected via a bus 5. The MPU 1 controls relative access to the flash memory 2 and comprises an empty sector management portion 4 which manages the empty sectors of the flash memory 2. The flash memory has the structure shown in figure 3 for example. The input-output unit 3 receives at the input data which are

**THIS PAGE BLANK (USPTO)**

to be stored in the flash memory 2 and delivers at the output the necessary data which have been read in the flash memory 2.

As will be seen in portion (A) of figure 3, the flash memory 2 consists of blocks B0 to Bm. Block B0 consists of sectors 00 to 0n, block B1 consists of sectors 10 to 1n, ..., and block Bm consists of sectors m0 to mn.

Each block corresponds to the minimum unit on which deletion can be carried out, that is where deletion can be deletion of the block type. Each block can correspond to a flash memory chip or to a portion of the flash memory chip. Consequently, the blocks B0 to Bm shown in portion (A) of figure 3 can correspond to one or more flash memory chips, and the flash memory 2 shown in figure 2 can consist of one or more flash memory chips.

The sectors forming each respective block have an arbitrary size which is selected in an appropriate manner. As described above with reference to figure 1, each sector comprises a logic address portion 10, a deletion management portion 11, a data portion 12 and an error detecting data portion 13. When the flash memory 2 is used as a file, for example, the data which are to be stored in the file are written in the data portion 12 of the sector.

As shown in portion (B) of figure 3, a deletion validating flag which indicates that the sector under consideration has been deleted and is invalidated is written to bits D7 and D6 of the deletion management portion 11 of this sector under consideration. Furthermore, a defective sector flag which indicates that the sector under consideration is defective is written to bits D5 and D4 of the deletion management portion 11 of this sector under consideration. In this embodiment, the deletion validating flag and the defective sector flag are written in double data groups to improve the reliability of the flash memory control apparatus, but it is obviously possible to provide more than two deletion validating flags

**THIS PAGE BLANK (USPTO)**

and more than two defective sector flags. The remaining bits, that is D3 to D0, of the deletion management portion 11 are reserve bits.

As can be seen in figure 3, if data are written to sector M0, the empty sector management portion 4 of the MPU 1 shown in figure 2 manages information indicating that the empty sectors begin from sector m1. If a read access demand is made from the input-output unit 3 destined for the flash memory 2 via the MPU 1, for example, the MPU 1 successively scans the logic addresses of each of the sectors contained within the flash memory unit from the sector 00 of portion (A) in figure 3. When the logic address of a sector in which the deletion management flag is not positioned agrees with the logic address of the destination of access, the data in this sector are read and transmitted to the input-output unit 3.

On the other hand, if a rewrite access demand is made from the input-output unit 3 destined for the flash memory 2, the MPU 1 determines in the flash memory 2 a sector having the logic address of the access destination and positions the deletion validating flag to bits D7 and D6 of the deletion management portion 11 of this sector. The flash memory 2 can store the datum "0" or "1" in the original state. The deletion validating flag is thus positioned at bits D7 and D6 of the deletion management portion 11 by positioning "1" if the bits D7 and D6 originally store "0" and positioning "0" if bits D7 and D6 initially store "1".

After this, the MPU 1 finds the sector from which the empty sectors begin in the empty sector management portion 4. For example, the MPU 1 determines sector m1 as being the sector from which the empty sectors begin and has access to the defective sector flag of bits D5 and D4 in the deletion management portion 11 of the sector m1. If the defective sector flag is not positioned in the deletion management portion 11 of the sector m1, the MPU 1 writes the logic address of the access destination in the logic address portion

**THIS PAGE BLANK (USPTO)**



10, the new data to be written in the data portion 12 and the error detection datum in the error detecting data portion 13 of the sector m1 and stores, in the empty sector management portion 4, information indicating that the next empty sector is sector m2.

However, if one of the bits D5 and D4 of the defective sector flag is positioned in the deletion management portion 11 of the sector m1, the MPU 1 will carry out the writing operation relative to the following section m2 in the manner indicated hereinbefore.

The write operation according to this embodiment will next be described in detail with reference to figure 4. Figure 4 is a flow chart showing the writing operation of the MPU 1 in figure 2.

As can be seen in figure 4, stage S1 introduces an input logic address originating from the input-output unit 3 from figure 2 and stage S2 scans the sectors of the flash memory 2 in which data have already been written. It is determined in stage S3 whether the flash memory 2 possesses a sector having the same logic address as the input logic address. The flow chart passes to stage S2 in the case of a positive response to stage S3 but advances to stage S6 in the case of a negative response to stage S3.

It is determined in stage S4 whether the sector having the same logic address as the input logic address has a deletion management portion 11 in which the deletion validating flag is positioned. The flow chart returns to stage S2 in the case of a positive response to stage S4. Conversely, if the response to stage S4 is negative, the deletion validating flag of the deletion management portion 11 of this sector is positioned in stage S5 and the datum which is to be written is introduced from the input-output unit 3 in stage S6. Next, the region of the flash memory 2 in which the datum is to be written is sought in stage S7.

**THIS PAGE BLANK (USPTO)**

In this case, the flash memory 2 has a total storage capacity attaining  $m-1$  blocks even if  $m$  blocks are provided, and one block is used as working block. As described hereinbefore, each block can correspond to one or more flash memory chips. Thus, from among the  $m$  flash memory blocks 2, only  $m-1$  blocks are used as the effective storage region of the flash memory 2.

In stage S8 it is determined whether or not the flash memory 2 comprises an empty region apart from the above-mentioned working block. The flow chart passes to stage S9 in the case of a negative response to stage S8 and passes to stage S12 in the case of a positive response.

The block having the greatest possible number of deletion validating flags positioned in the sectors of the blocks is sought in stage S9. Those of the sectors forming the block found in stage S9 in which the deletion validating flag is not positioned are then saved in the working block in stage S10 by copying the content of the sectors in the working block. After this, the block found in stage S9 is deleted in stage S11 and is then considered as the new working block.

Stage S12 is intended to confirm that the defective sector flag of the sector placed at the beginning of the empty region is not positioned and passes to the following sector if the defective sector flag is positioned. The address, the data and the error detecting datum are written in the empty region in stage S13.

In stage S14 it is determined whether or not the write operation has succeeded. The flow chart ends in the case of a positive response to stage S14. Conversely, in the case of a negative response, it proceeds to stage S15, the positioning of the defective sector flag for the sector relative to which the write operation has not succeeded, then the flow chart returns to stage S8.

**THIS PAGE BLANK (USPTO)**

Once the write operation relating to the storage region, that is to m-1 blocks, has ended, it is possible to adjust the data by means of the working block. For example, the block (or storage memory chip) having the greatest deletable region is determined from among the blocks (or the chip or the flash memory chip) in which the data have already been written. In the block which has the greatest deletable region, the region which is not to be deleted is saved in the working block by copying its content, and the datum which is to be written next is written after this content. Furthermore, the above-mentioned block possessing the greatest deletable region is deleted once the content to be saved has been copied in the working block, and the deleted block is prepared to be used as new working block during the adjustment of following data.

The read operation according to this embodiment will now be described in detail. Figure 5 is a flow chart illustrating the read operation of the MPU 1 shown in figure 2.

As can be seen in figure 5, stage S21 introduces an input logic address originating from the input-output unit 3 in figure 2, and stage S22 scans the sectors of the flash memory 2 in which data are already written. It is determined in stage S23 whether the flash memory 2 possesses a sector having the same logic address as the input logic address. The flow chart passes to stage S25 in the case of a positive response to stage S23 but goes to stage S24 in the case of a negative response to stage S23. A read error is indicated to the input-output unit 3 via the MPU 1 in stage S24.

It is decided in stage S25 whether the sector having the same logic address as the input logic address possesses a deletion management portion 11 in which the deletion validating flag is positioned. The flow chart returns to stage S22 in the case of a positive response to stage S25. Conversely, in the case of a negative response to stage S25, the data of the sector having the same logic address as the

**THIS PAGE BLANK (USPTO)**

input logic address are read in stage S26 and the data read are transmitted to the input-output unit 3 in stage S27.

An error detecting datum which is associated with the error read and indicates the error located there is produced in stage S28. In stage S29, the error detecting datum produced in stage S28 is compared with the error detecting datum coming from the error detecting data portion 13 of the sector. It is determined in stage S30 whether or not the two compared error detecting data agree. The flow chart ends in the case of a positive response to stage S30 but passes to stage S31 in the case of a negative response. A read error is indicated to the input-output unit 3 via the MPU 1 in stage S31.

It is also determined in stage S32 whether or not the defective sector flag contained in the deletion management portion 11 of the sector is positioned. The flow chart ends in the case of a positive response to stage S32. Conversely, the defective sector flag of the deletion management portion of the sector is positioned in stage S33 then the flow chart ends in the case of a negative response.

On the basis of the process and apparatus which have just been described by way of non-limiting examples, a skilled person can obviously imagine different variations and modifications not departing from the scope of the invention.

**THIS PAGE BLANK (USPTO)**



## Claims

1. Process for flash memory control, characterised in that the following operations are proposed:

(a) dividing a storage region of a flash memory (2) into several sectors each containing a logic address portion (10) intended to store a logic address of the sector, a deletion management portion (11) intended to store information indicating at least whether the sector can be deleted, and a data portion (12) intended to store data; and

(b) having access to an arbitrary sector of the flash memory (2) whilst specifying the logic address of the arbitrary sector.

2. Process according to claim 1, characterised in that the storage region of the flash memory (2) is a block type deletable region where each block is the minimum unit for which deletion of the flash memory can be carried out, and said operation (a) divides the storage region in such a way that each block consists of several sectors.

3. Process according to claim 2, characterised in that the flash memory (2) consists of one or more flash memory chips.

4. Process according to claim 2, characterised in that the total storage region of the flash memory (2) amounts to  $m$  blocks, and  $m-1$  blocks are employed as an effective storage region whereas the remaining block is used as a working block.

5. Process according to claim 4, characterised in that the following operation is also proposed:

(c) using the working block to adjust the data which have been written in the flash memory (2) after a write operation relating to all the  $m-1$  blocks has ended.

**THIS PAGE BLANK (USPTO)**

6. Process according to claim 1, characterised in that operation (a) divides the storage region in such a way that each sector also comprises an error detecting data portion (13) intended to store information which is used to detect and correct an error in the data stored in the data portion (12).

7. Process according to claim 1, characterised in that operation (a) divides the storage region in such a way that the deletion management portion (11) of each sector comprises a deletion validating flag which possesses an original state if the data portion (12) of this same sector is valid and possesses a state different from the original state if the data portion of this sector has been invalidated.

8. Process according to claim 7, characterised in that the deletion management portion (11) of each sector comprises several deletion validating flags.

9. Process according to claim 1, characterised in that operation (a) divides the storage region so the deletion management portion (11) of each sector comprises a defective sector flag which indicates whether or not the sector is defective.

10. Process according to claim 9, characterised in that the deletion management portion (11) of each sector comprises several defective sector flags.

11. Process according to claim 1, characterised in that the following operation is also proposed: (c) managing the empty sectors of the flash memory (2) on the basis of the information stored in the deletion management portion (11) of each sector.

**THIS PAGE BLANK (USPTO)**

12. Apparatus for flash memory control, characterised in that the following are provided:

a flash memory (2) possessing a storage region which is divided into several sectors each comprising a logic address portion (10) intended to store a logic address of the sector, a deletion management portion (11) serving to store information indicating at least whether the sector can be deleted, and a data portion (12) intended to store data; and

a control means (1) coupled to said flash memory in order to have access to an arbitrary sector of said memory (2) by specification of the logic address of the arbitrary sector.

13. Apparatus according to claim 12, characterised in that the storage region of said flash memory (2) is a block type deletable region, each block is the minimum unit for which deletion of the memory (2) can take place and each block consists of several sectors.

14. Apparatus according to claim 13, characterised in that said flash memory (2) consists of one or more flash memory chips.

15. Apparatus according to claim 13, characterised in that the total storage region of said flash memory (2) amounts to  $m$  blocks, and  $m-1$  blocks are used as effective storage region whereas the remaining block forms a working block.

16. Apparatus according to claim 15, characterised in that said control means (1) comprises a means for adjusting, by means of the working block, the data which have been written in said flash memory (2) after a write operation relating to all the  $m-1$  blocks has ended.

**THIS PAGE BLANK (USPTO)**

17. Apparatus according to claim 12, characterised in that each sector also comprises an error detecting data portion (13) intended to store information which is used to detect and correct an error in the data stored in the data portion (12).

18. Apparatus according to claim 12, characterised in that the deletion management portion (11) of each sector comprises a deletion validating flag which possesses an original state if the data portion of this same sector is valid and possesses a state other than the original state if the data portion of this sector is invalidated.

19. Apparatus according to claim 18, characterised in that the deletion management portion (11) of each sector comprises several deletion validating flags.

20. Apparatus according to claim 12, characterised in that the deletion management portion (11) of each sector comprises a defective sector flag which indicates whether or not the sector is defective.

21. Apparatus according to claim 20, characterised in that the deletion management portion (11) of each sector comprises several defective sector flags.

22. Apparatus according to claim 12, characterised in that the said control means (1) comprises a means (4) serving to manage the empty sectors of said flash memory (2) on the basis of the information stored in the deletion management portion (11) of each sector.

**THIS PAGE BLANK (USPTO)**



## Captions

### Figure 1

bloc = block  
 secteur = sector  
 10 logic address  
 11 deletion management  
 12 data  
 13 error detection datum

### Figure 2

4 empty sector management  
 2 flash memory

### Figure 3

(A) flash memory unit  
     sector  
 10 logic address  
 11 deletion management  
 12 data  
 13 error detection datum  
 (B) deletion validating flag  
     defective sector flag

### Figure 4

start  
 S1 introduce address  
 S2 scan sectors  
 S3 addresses agree?  
     oui = yes  
     non = no  
 S4 deletion validating flag positioned?  
 S5 position deletion validating flag

**THIS PAGE BLANK (USPTO)**

S6 introduce data originating from unit 3  
 S7 seek data read region  
 S8 region empty apart from working block?  
 S9 seek block having greatest number of deletion validating flags  
 S10 save sectors not to be deleted  
 S11 delete block  
 S12 confirm that that defective sector flag is not positioned  
 S13 write the address, data, etc. in the empty region  
 S14 write operation successful?  
 S15 position the defective sector flag in the sector where the write operation was not successful  
 end

Figure 5

start  
 S21 introduce address  
 S22 scan sectors  
 S23 addresses agree?  
 S24 indicate read error to unit 3  
 S25 write validating flag positioned?  
 S26 read data  
 S27 transmit data to unit 3  
 S28 produce error detecting datum  
 S29 compare error detecting data  
 S30 compared data identical?  
 S31 indicate read error to unit 3  
 S32 defective sector flag positioned?  
 S33 position defective sector flag  
 end

**THIS PAGE BLANK (USPTO)**